

CS3025 Compiladores

Laboratorio 5.1 – 15 Septiembre 2023

Esta semana empezamos a estudiar el Visitor Pattern.

El Patron Visitor nos permite implementar distintas interpretaciones de ASTs eficientemente. Cada interpretación esta asociada a un visitor, el cual debe implementar la clase abstracta Visitor (o similar). De esta manera el código asociado a cada interpretación (análisis semántico, generación de código) estará asociado a un visitor diferente, en un archivo diferente.

El plan es el siguiente:

- Leer, compilar y correr el código que implementa la impresión y evaluación de expresiones aritmeticas. La implementación sigue el estilo orientado a objetos.
- Implementar impresión y evaluación de expresiones aritmeticas usando visitors.
- Implementar impresiones y evaluación de programa IMP usando visitors.

Primero, bajar y expandir el archivo lab.5.1.zip. Esto generara 3 folders: `exp`, `exp_ast` e `imp_ast`.

1. Un evaluador de expresiones aritmeticas: folder `exp`

En las preguntas 1 y 2 estaremos trabajando con expresiones aritméticas definidas por la siguiente sintaxis abstracta:

```
Exp ::= Exp (('+' | '-' | '*' | '/') Exp | num | '(' Exp ')'
```

El folder `exp` contiene la implementación de expresiones aritméticas y distintas interpretaciones. El programa final puede compilarse con

```
>>> g++ exp_test.cpp aexp_parser.cpp exp.cpp
```

Y ejecutarse de la siguiente manera:

```
>>> ./a.out "2**3 + 34"
expression: 2 ** 3 + 34
eval 42
```

2. Visitors para Exp: folder `exp_ast`

El folder `exp_ast` contiene los archivos necesarios para implementar visitantes para imprimir y evaluar (por implementar) expresiones aritméticas. El programa `exp_ast_test.cpp` imprime expresiones aritméticas utilizando `ASTPrinter`, un visitante que implementa los métodos definidos por la clase abstracta `ASTVisitor`. Las clases en `exp_ast.hh` contienen los métodos `accept` del Visitor Pattern – esto se hace una vez. Podemos compilar y ejecutar el programa de la siguiente manera:

```
>>> g++ exp_ast_test.cpp aexp_parser.cpp exp_ast.cpp ast_printer.cpp
>>> a.out "2**3 + 34"
Expression : 2 ** 3 + 34
```

Las implementaciones del evaluador de expresiones aritméticas esta incompleta. Completar los archivos `ast_evaluator.hh` y `ast_evaluator.cpp`, y quitar los comentarios a `exp_ast_test.cpp` para probar el nuevo código. Compilar de la siguiente manera:

```
>>> g++ exp_ast_test.cpp aexp_parser.cpp exp_ast.cpp ast_printer.cpp
ast_evaluator.cpp
```

3. Visitors para IMP: folder `imp_ast`

El folder `imp_ast` contiene los archivos necesarios para implementar visitantes para imprimir y evaluar (por implementar) programas IMP. Solo `ImpPrinter` esta implementado. Compilar así:

```
>>> g++ imp_ast_test.cpp imp_parser.cpp imp_ast.cpp imp_printer.cpp
>>> ./a.out "x = 10; print(x** 2)"
Program : {
  x = 10;
  print(x ** 2);
}
```

Las implementaciones del interprete esta incompleta – pero nótese como se ha movido el campo memoria a esta clase.. Completar los archivos `imp_interpreter.hh` e `imp_interpreter.cpp`, y quitar los comentarios a `imp_ast_test.cpp` para probar el nuevo código. Compilar de la siguiente manera:

```
>>> g++ imp_ast_test.cpp imp_parser.cpp imp_ast.cpp imp_printer.cpp
imp_interpreter.cpp
```